

卒業研究報告題目

Sentence-BERTを用いた
レビューの極性判定の研究

Research on Sentiment Classification of Reviews
Using Sentence-BERT

指導教員 出口利憲 教授

岐阜工業高等専門学校 電気情報工学科

2021E05 伊縫海人

令和8年(2026年) 2月12日提出

Abstract

In recent years, AI has begun to be used in various fields such as image analysis, dialogue with people, and text analysis. The purpose of this study is to find ways to improve the classification accuracy of AI. Therefore, we created an AI to detect whether opinions in game reviews are positive or negative, varied different factors in the learning process, and examined the accuracy of the classification. As a result, we found that when performing classification using AI, it is effective to have a large amount of data and few training sessions, and that if it is known in advance that there will be a difference in the amount of positive and negative opinions, it is effective to use a method that emphasizes the characteristics of minority opinions.

目次

Abstract	i
第1章 序論	1
第2章 ニューラルネットワーク	2
2.1 ニューロン	2
2.2 ニューラルネットワーク	2
2.3 学習方法	2
2.3.1 勾配降下法	2
2.3.2 ドロップアウト	5
2.3.3 誤差逆伝播法	5
第3章 活性化関数	6
3.1 活性化関数	6
3.2 活性化関数の種類	6
3.2.1 Step 関数	6
3.2.2 Sigmoid 関数	6
3.2.3 ReLU 関数	8
3.2.4 Softmax 関数	8
3.2.5 恒等関数	9
第4章 テキストマイニング	10
4.1 テキストマイニング	10
4.2 分析手法	10
4.2.1 センチメント分析	10
4.2.2 共起分析	10
4.2.3 対応分析	11
4.2.4 主成分分析	11
第5章 分類の評価	12
5.1 分類における精度	12
5.2 評価指標	12
5.2.1 正答率 (Accuracy)	12

5.2.2	適合率 (Precision)	12
5.2.3	再現率 (Recall)	13
5.2.4	F1 スコア	13
5.2.5	AUC	13
第 6 章	ライブラリ	15
6.1	BERT	15
6.2	Sentence-BERT	15
6.3	PyTorch	15
第 7 章	モデル解説	17
7.1	データセット	17
7.2	レビューのデータ化	17
7.3	極性判定モデルの構成	17
7.4	学習	18
第 8 章	実験	20
8.1	特徴の表現	20
8.2	データ量の違い	22
8.3	データ量の差の解消	26
8.4	文章分割による違い	31
第 9 章	結論	34
	参考文献	35

第1章 序論

近年 AI の登場により従来の人がルールを記載する手法ではなくデータからルールを学習させる手法が可能になり、様々な製品に様々な用途で AI が使用されるようになった。例えば画像の特徴量を学習することにより入力した画像が何の画像なのか判別したりするデータの解析、利用者が依頼した通りの画像や画像からアニメーションを生成したりする画像やアニメーションの生成、人とコミュニケーションを行ったりする人と対話を行う AI などが登場した。また、AI により大量のデータを人の手より短時間で処理できるようになり、いまや AI は人類においてとても身近な存在であり人を支えるパートナーともいえる存在となった。

しかし、AI も完璧な存在ではない。AI の解析精度は学習データに強く依存するため AI の判断結果と人から見た判断結果が異なることが発生しやすい。そのためこの研究では人類が AI を使用するにおいて多く活用するテキストデータの解析を行うにあたってどのような要素が重要かを調べるため、自分自身が興味のあるゲームのレビューの極性判定を正確に行うために学習に必要な様々な要素を変化させることにより調べた。

第2章 ニューラルネットワーク

2.1 ニューロン

ニューロンとは脳の中で情報を受け取り次の細胞や器官へ情報の伝達を行う役割を果たしている細胞である。ニューロンは Figure 2.1 で示す形をしており、細胞本体である「細胞体」、情報を受け取る「樹状突起」、情報を伝える「軸索」、情報を入力する「シナプス」から構成されており人間の脳の場合この細胞が 100 億から 1000 億個存在していると言われている¹⁾。

2.2 ニューラルネットワーク

ニューラルネットワークとはニューロンの仕組みを模した計算モデルであり、その特徴として複雑なデータの関係性を捉えることができ、画像認識や音声認識、自然言語の処理などの多岐にわたる分野で活用されている。ニューラルネットワークの構造は Figure 2.2 のように入力層、隠れ層、出力層に分かれている。入力層はデータを受け取る役割、隠れ層はデータの特徴量を抽出する役割、出力層は最終的な予想や分類の結果を出力する役割を果たしている。入力層と出力層は 1 つしか存在できないものの隠れ層は複数存在することができ、隠れ層を多層化することでより表現力を高めるディープニューラルネットワークを作成することができる²⁾。

2.3 学習方法

ニューラルネットワークの学習方法は教師あり学習と教師なし学習の二種類がある。教師あり学習はデータ学習にラベルを付与することでそのラベルにはどのような特徴があるのかを学習する手法である。例えば画像認識でその画像はどの動物の画像かを認識する場合ラベルにはどの動物かを示す番号や名前を使用する。教師なし学習は正解データがない学習方法であり、入力したデータが持つ頻出パターンの発見や分類に適している。

2.3.1 勾配降下法

勾配降下法とは数学的最適化の一つであり、関数の極値を見つける手法である。深層学習において予測の間違いを最小限に抑えるパラメーターを求める手法として勾配降下法を使用することで予測の間違いを減らすために利用される。学習率を η 、損失関数を

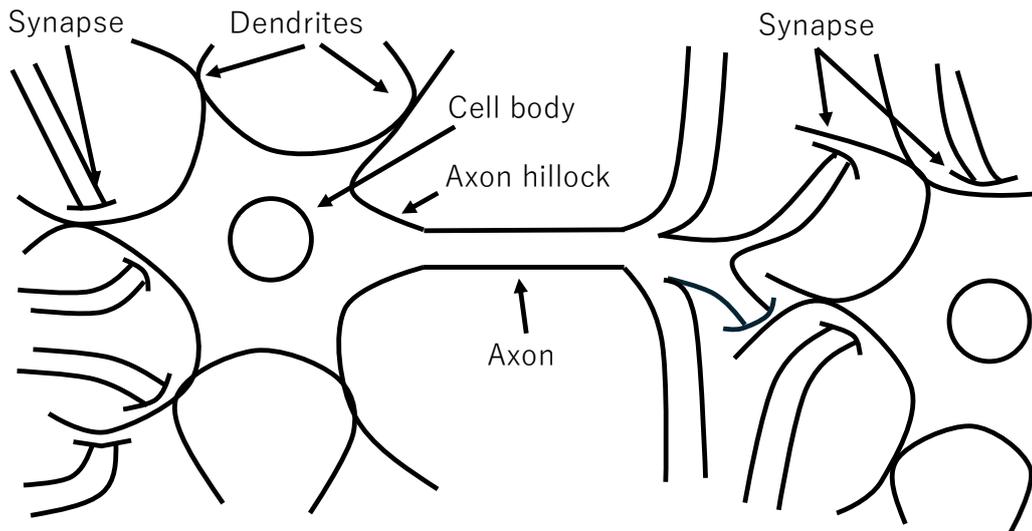


Figure 2.1 Neuron.

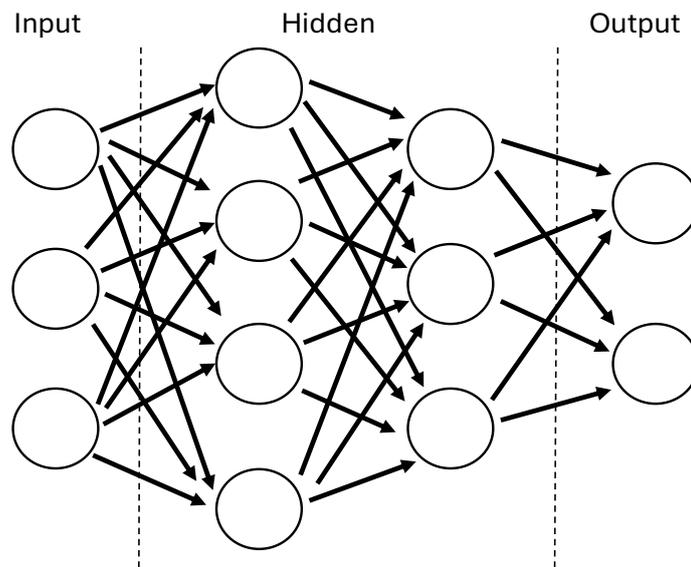


Figure 2.2 Neural Network.

L 、学習のステップを n として表すとパーセプトロンの重み w は式 (2.1)、バイアス b は式 (2.2) により更新できる。

$$w_{n+1} = w_n - \eta \nabla L(w) \quad (2.1)$$

$$b_{n+1} = b_n - \eta \nabla L(b) \quad (2.2)$$

更新の推移は学習率によって変化するため、関数 $y = x^2$ を例にとって考えると学習率が低い場合は Figure 2.3 の左のように推移し、学習率が高い場合は Figure 2.4 のように推移する³⁾。

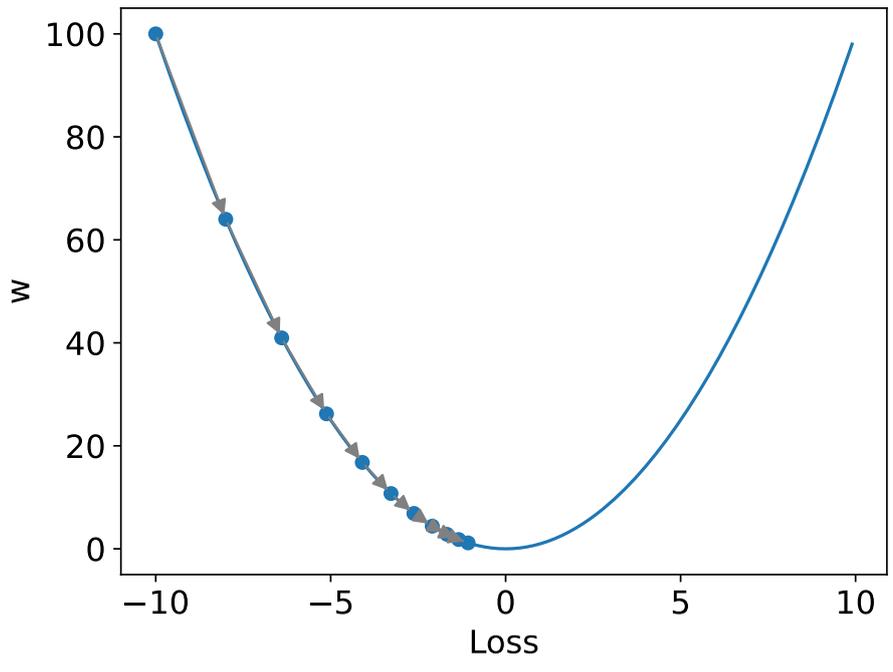


Figure 2.3 Low Learning Rate.

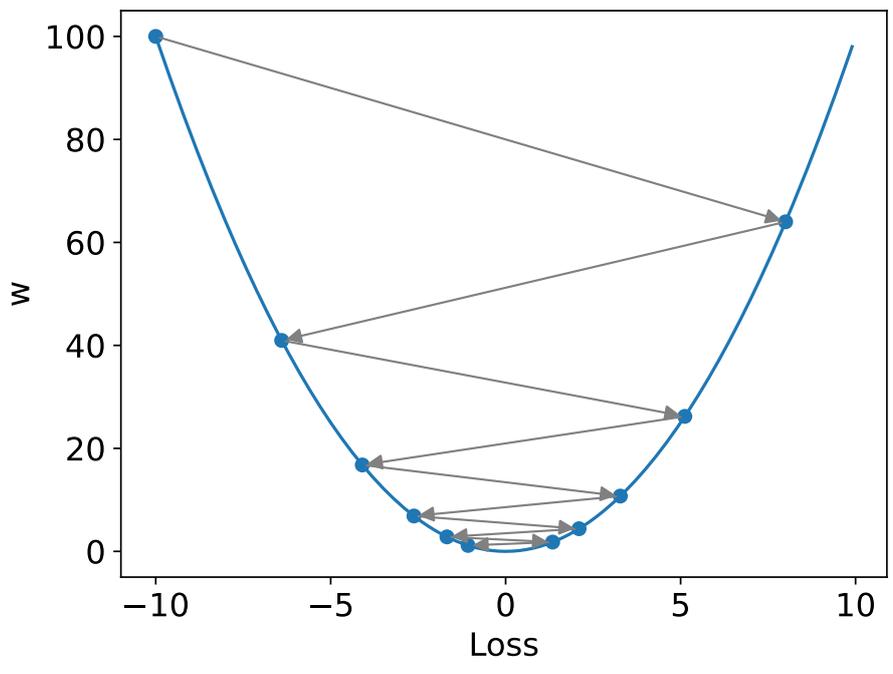


Figure 2.4 High Learning Rate.

2.3.2 ドロップアウト

ドロップアウトとは人の忘却を学習時に一定割合のノードを不活性させることで再現している。ドロップアウトを使用することでニューロンの共依存関係の解消、擬似的なアンサンブル学習の効果、特徴量の多様化による未知のデータへの対応力を高めることができる。ニューラルネットワークの規模が大きいほどパラメータ数が多く過学習が起きやすいため、ドロップアウトは有効な過学習対策となる。しかし、ドロップアウト率を適切に設定しなければドロップアウトの効果が発生しない、特徴を全く学習できなくなる等の問題が発生してしまう点に注意が必要である⁴⁾。

2.3.3 誤差逆伝播法

誤差逆伝播法とは出力と正解データの差をネットワークの各層に遡って伝播させるアルゴリズムであり、誤差を最小化すること、効率的に学習を行うこと、深層ネットワークの最適化を行うことができる仕組みとして入力データから計算を行いその出力を算出した後正解データと出力データの誤差を計算する。誤差を計算する関数は損失関数と呼ばれ、平均二乗誤差は回帰問題で、クロスエントロピーは分類問題で使われる。誤差をネットワーク出力から入力に向かって伝播させることでニューロンの重みを計算する。勾配降下法などの最適化手法を用いてネットワークのパラメータを更新する。このとき、損失関数を各パラメータで偏微分することで勾配を求め、その勾配方向に基づいてパラメータ更新を行う。

第3章 活性化関数

3.1 活性化関数

活性化関数とは次のニューロンにどのような情報を入力するかを決める役割を持っており、ニューラルネットワークにおいて難しい予測や分類をするために必要な要素となっている。活性化関数があることによって分類をするための境界線が直線ではなく曲線でも表現できるようになる。

3.2 活性化関数の種類

活性化関数は Step 関数、シグモイド関数、ReLU 関数、Softmax 関数、恒等関数の 5 種類ある。

3.2.1 Step 関数

Step 関数は入力された値が 0 以下であれば 0 を出力、それ以外であれば 1 を出力する関数であり式 (3.1) で表すことができる。

$$f(x) = \begin{cases} 0 & (x \leq 0) \\ 1 & (x > 0) \end{cases} \quad (3.1)$$

Step 関数をグラフとして表現したものが Figure 3.1 となる。

3.2.2 Sigmoid 関数

Sigmoid 関数は入力された値を 0 から 1 の値に変換する関数であり、式 (3.2) で表すことができる。

$$f(x) = \frac{1}{1 + \exp(-x)} \quad (3.2)$$

また、Sigmoid 関数をグラフとして表現したものが Figure 3.2 となる。Sigmoid 関数は数値を 0 から 1 に変換するため 2 クラスの分類問題に使用できる。

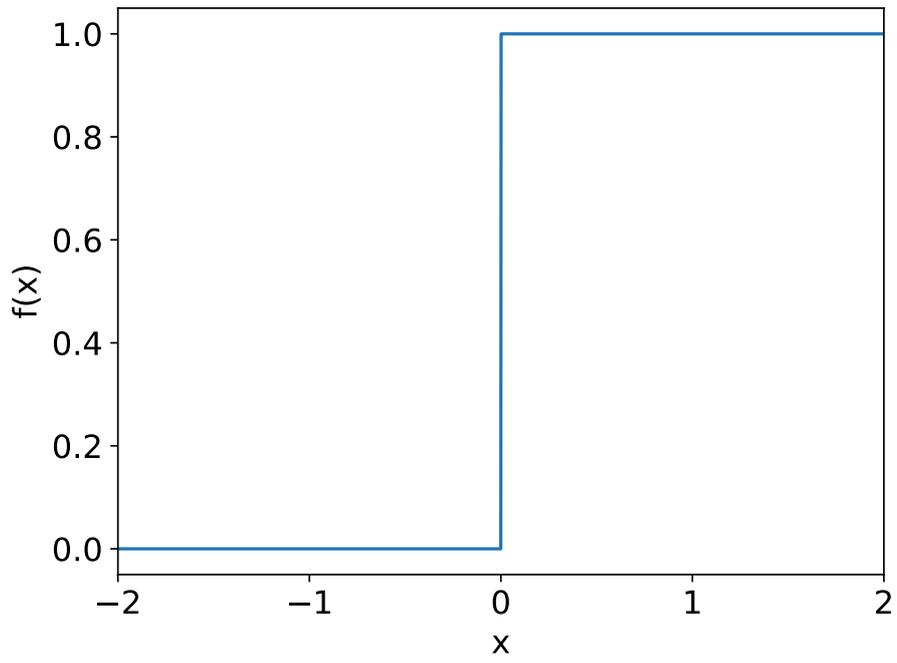


Figure 3.1 Step function.

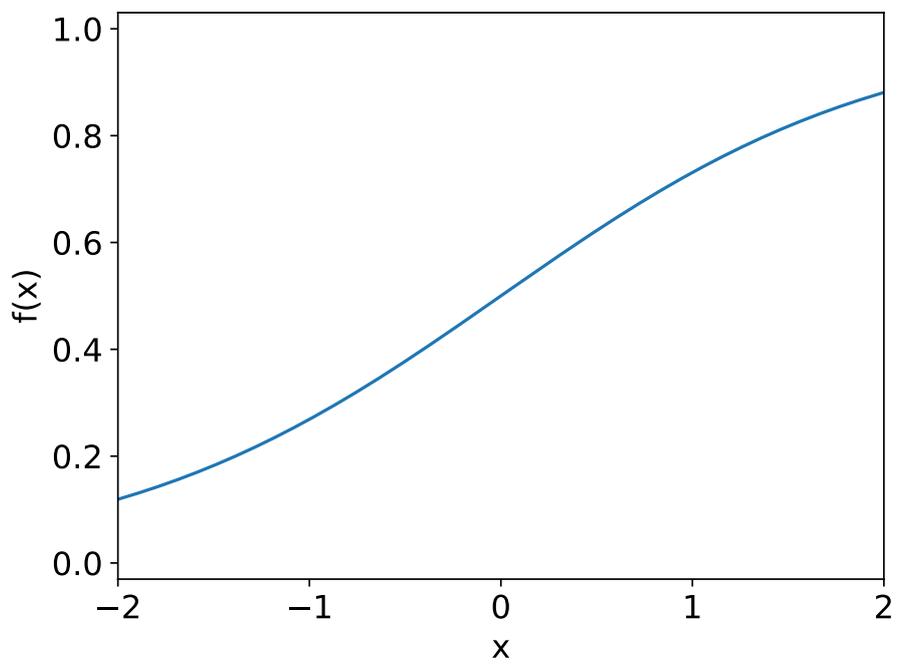


Figure 3.2 Sigmoid function.

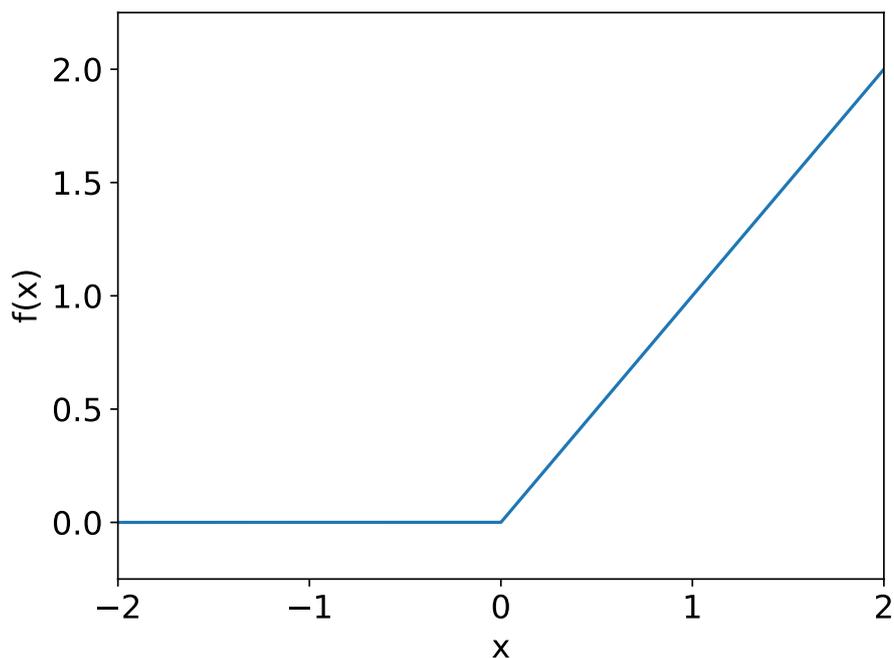


Figure 3.3 ReLU function.

3.2.3 ReLU 関数

ReLU 関数は入力された値がマイナスであれば0として出力し、それ以外はその値をそのまま出力する関数であり、式 (3.3) で表すことができる。

$$f(x) = \begin{cases} 0 & (x \leq 0) \\ x & (x > 0) \end{cases} \quad (3.3)$$

また、ReLU 関数をグラフとして表現すると Figure 3.3 となる。

3.2.4 Softmax 関数

Softmax 関数とは出力結果を割合として表す関数であり、出力層に n 個のノードが存在するとき、出力層の i 番目の出力値 x_i に対する Softmax 後の確率値 $f_i(x)$ は式 (3.4) で表すことができる。

$$f_i(x) = \frac{e^{x_i}}{\sum_{k=1}^n e^{x_k}} \quad (3.4)$$

Softmax 関数は出力を割合で表すことからデータがどのクラスに属するかを判断する分類問題に多く使用されている。また、sigmoid 関数との運用の違いとして sigmoid 関数は

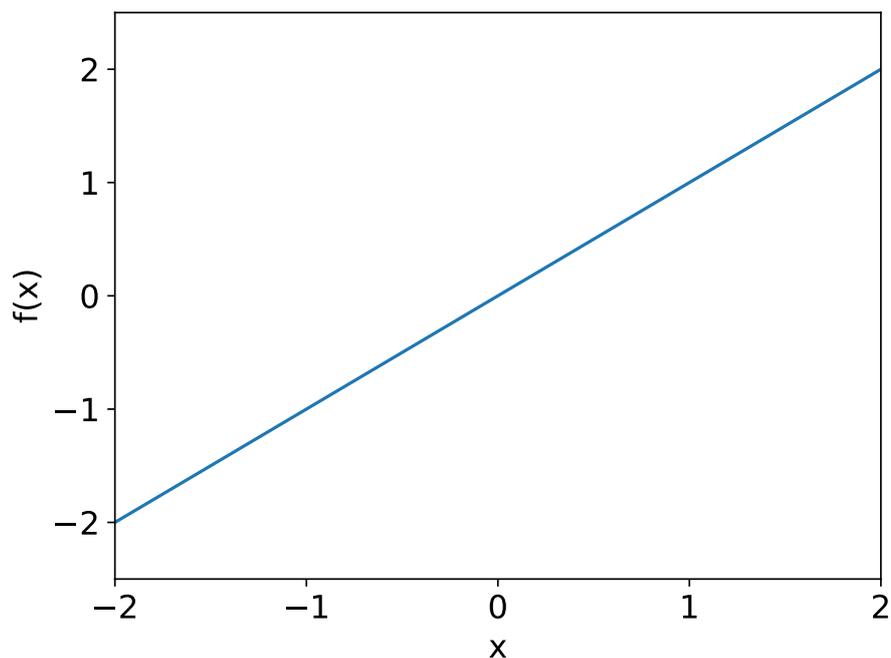


Figure 3.4 Identity function.

クラスが2つ存在する場合の分類問題、softmax関数はクラスが3つ以上の分類問題に使用される点がある⁶⁾。

3.2.5 恒等関数

恒等関数は入力された値をそのまま返す関数であり、式(3.5)で表すことができる。

$$f(x) = x \quad (3.5)$$

また、恒等関数をグラフとして表現するとFigure 3.4となる。恒等関数は入力された数値を返すことから回帰問題において予測した数値を出力として渡すために使われている⁵⁾。

第4章 テキストマイニング

4.1 テキストマイニング

テキストマイニングとは自由な形式で記述された文章を、自然言語処理の技術で単語や文節に分割して出現頻度や相関関係といった情報から有益な情報を探し出す技術のことである。テキストマイニングを活用することで以前はテキストデータを人の目を通さなければならなかったがコンピュータ上でも行えるようになり大量のデータを素早く処理できるようになった⁷⁾。テキストマイニングは自然言語処理で文章を単語毎に分割する、ルールやパターンをもとに辞書を整理する、単語の重要度や関連度や感情などを分析する、必要な情報を整理された形式で出力するという流れで行われる⁸⁾。テキストマイニングにはデータの特徴や傾向を見つけることができる探索的データ解析と、テキストデータの特徴を抽出しカテゴリごとに分類する文書分類がある⁹⁾。

4.2 分析手法

テキストマイニングには感情を読み取るセンチメント分析、関連性を可視化する対応分析と主成分分析、複数の単語の結びつきを探る共起分析などの多様な手法が存在する。

4.2.1 センチメント分析

センチメント分析はテキストマイニングの代表的な手法であり、文を「肯定的」「中立的」「否定的」の三つで評価する手法である。この手法は肯定的とも否定的ともとれる単語や文に弱いことであり世代や文脈によって解釈が異なる場合に正しく判定しにくい。

4.2.2 共起分析

共起分析は文章中に存在する二つの単語が同時に使用されている度合いをもとに解析を行う。例えば「曲」という単語に対して「美しい」「不快」という単語を組み合わせることで「曲」と同時に出てくる単語の共起率を調べることで曲に対してどのような感想を持っているか読み取ることができる。

4.2.3 対応分析

対応分析はコレスポネンス分析とも呼ばれており単語間の関係性を散布図で視覚的に表現する手法でありデータ構造や類似性を視覚的に把握したいとき、集計項目と分類項目の両方が多い場合に有効である。

4.2.4 主成分分析

主成分分析とは多くの変数を持つデータを集約して主成分を作成する統計的分析手法であり、文章内にどの要素がどれ程含まれているかを判別することやデータの項目を削減することに使用できる¹⁰⁾。

第5章 分類の評価

5.1 分類における精度

分類における精度とは分類がどれ程正確に行えているかを表す指標である。精度を計算する上で必要になる要素は真陽性 (TP)、真陰性 (TN)、偽陽性 (FP)、偽陰性 (FN) である。真陽性は正しく陽性と判断できたもの、真陰性は正しく陰性と判断できたもの、偽陽性とは誤って陽性と判断したもの、偽陰性は誤って陰性と判断したものである。つまり、正しく判定をできたことを示す真陽性と真陰性というデータと、誤った判定を出したことを示す偽陰性、偽陽性が必要となる。

5.2 評価指標

精度の評価指標として正答率、適合率、再現率、F1 スコア、AUC が存在する。

5.2.1 正答率 (Accuracy)

正答率 (Accuracy) は全ての予測のうち正解した割合を示す評価指標であり、式 (5.1) で求めることができる。

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.1)$$

精度指標の中では最も分かりやすいが学習データに大きな偏りがある場合信用できない評価指標になる。例えば学習データのほとんどが陽性だった場合と陽性と陰性の比率が同じだった時全てを陽性と判別するとほとんどが陽性だった場合の正答率は高くなり陰性を正しく判別できているかわからなくなるため判別する際はデータに偏りがないようにする必要がある。

5.2.2 適合率 (Precision)

適合率 (Precision) は分類において陽性と判断したもののうち実際に陽性だったものの割合を示す評価指標であり、式 (5.2) で求めることができる。

$$Precision = \frac{TP}{TP + FP} \quad (5.2)$$

適合率が高ければ高いほど陰性の誤検知が少ないことを表しているため、陰性の誤検知

がどの程度存在するかがわかるが陽性を見逃しがどの程度かわからないという欠点が存在する。

5.2.3 再現率 (Recall)

再現率は分類において実際に陽性だったものをどれだけ見つけることができたかを表す評価指数であり、式 (5.3) で求めることができる。

$$Recall = \frac{TP}{TP + FN} \quad (5.3)$$

再現率が高いほど陽性を見逃しが少ないことを表しており陽性の検知を重要視する場合有効となる指標である。再現率は適合率と反対に陽性を見逃しがどの程度かわかるメリットが存在する代わりに陰性の誤検知がどの程度かわからないデメリットが存在する。再現率は適合率と対になる評価指数であるため、再現率を上げようとするとう適合率が下がってしまい、適合率を上げようとするとう再現率が下がってしまう。

5.2.4 F1 スコア

F1 スコアは適合率と再現率の両方を考慮した場合の評価基準であり、式 (5.4) で求めることができる。

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5.4)$$

式 (5.4) の分子で掛け算を行っているため、適合率か再現率のどちらかが極端に低いと F1 スコアも低くなる特徴がある。F1 スコアが最も高くなる場合が適合率と再現率のバランスが最も取れた状態であるといえる。F1 スコアは学習データに大きな偏りがある場合でも使用できる評価指数であるメリットが存在するが、学習データに偏りがある場合は正答率と近い値をとるため計算するメリットが少なくなるデメリットがある。

5.2.5 AUC

AUC とは ROC 曲線と PR 曲線の下面積を表す言葉である。ROC 曲線とは分類において全ての不良品のうち不良品と判断したものの割合である真陽性率を縦軸、全ての良品のうち不良品と判断したものの割合である偽陽性率を横軸としてプロットした曲線である。PR 曲線とは縦軸に適合率、横軸に再現率をプロットした曲線である。ROC 曲線

と PR 曲線による AUC の違いとして ROC は偏りの小さいデータの場合全体的な精度を掴むことができ、PR 曲線ではデータの偏りが発生した場合でも使用できるという特徴がある¹¹⁾。

第6章 ライブラリ

6.1 BERT

BERTとはGoogle社のJacob Devlin氏らが発表したBidirectional Encoder Representations from Transformersと呼ばれる自然言語処理モデルを略したものである。名前の通りTransformerによる双方向のエンコード表現、つまり文章を文頭、文末から学習する特徴があるため文脈を読むことができる。BERTはすでに入力されたシーケンスを元に別のシーケンスを予測する事前学習モデルであり、ラベルが付与されていない分散表現をTransformerがMasked Language ModelとNext Sentence Predictionという二つの手法を同時進行で進めることで学習するという仕組みである¹²⁾。

6.2 Sentence-BERT

Sentence-BERTとはBERTの一種であり、BERTが苦手とする複数の文章のクラスタリングを可能にしたモデルである。Sentence-BERTの構造はFigure 6.1となる。ファインチューニングする際の目的関数はラベルによって変化する。ラベルが正誤であればFigure 6.1の左のように文章ベクトルを連結して分類問題として解く、ラベルが数値であればFigure 6.1の右のようにコサイン類似度を算出する¹³⁾。

6.3 PyTorch

PyTorchとはMetaが中心となって開発・公開したオープンソースのディープラーニングフレームワークである。PyTorchには「動的グラフである点」、「Pythonicなコード記法」、「豊富な拡張ライブラリ群」、「研究から実運用までのシームレスな移行」というメリットがあり下記はメリットについての詳細な説明である。「動的グラフである点」については動的にグラフを構築する特徴から複雑なモデルアーキテクチャや条件分岐を直感的に実装でき、中間層の出力を確認することでデバッグやモデル改良が容易に行える。「Pythonicなコード記法」についてはPythonで完結する上、Numpyに似たテンソル操作を行えるため直感的なモデル定義やトレーニンググループの記述が可能であり学習コストが低い。「豊富な拡張ライブラリ群」については画像処理用のTorchVision、自然言語処理用のTorchText、オーディオ分析用のTorchAudioなどの豊富なライブラリがあり、多くの課題に対処しやすい。「研究から実運用までのシームレスな移行」について、

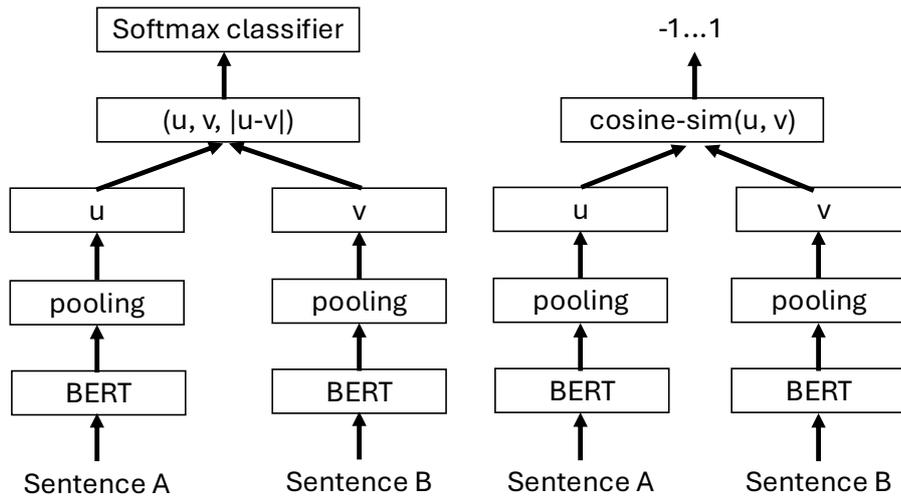


Figure 6.1 Sentence-BERT.

PyTorch は TorchScript や ONNX 形式への変換が可能であり、PyTorch Lightning を使用することで実装スピードと可読性を高めることもできる。本実験では学習を行っている過程で AI の性能はどのように変化しているかを調べるため PyTorch を使用した¹⁴⁾。

第7章 モデル解説

今回の実験ではゲームの商品レビューを学習し、未知なるテキスト文に対してそのレビュー文はどのような評価をしているかを極性判定するAIを作成した。

7.1 データセット

データセットであるゲームのレビューとそのレビューが肯定的な意見か否定的な意見かを判別するラベルはSteamから収集する。任意のゲームの中から最新レビュー順に40ページ分日本語のレビューに限定しjson形式で収集する。本実験で収集したゲームと集まったゲームレビューの数をTable 7.1に示す。得たデータセットは肯定的な意見が多いため、肯定的な意見と否定的な意見を同じ件数に調整した結果、データ数は22,856件となった。なお、本研究では計算量削減および学習効率を考慮し、得られたデータセットの中から肯定意見と否定意見の比率が変動しないように一部を抽出し、学習データを10,000件、検証用データを5,714件として使用した。

7.2 レビューのデータ化

収集したレビューは文章内に含まれる「。、!、?」で区切ることでレビューを文の集合に変換する。その後各文に対してSentence-BERT系モデルであるSentenceTransformer(”sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2”)を使用し、文章の意味情報を保持した数値ベクトルへ変換する。それぞれの文を変換した数値ベクトルは384次元であり、数値ベクトルの集合に対して平均プーリングと最大プーリングを適用し、これらを連結することでレビュー全体を表現する768次元ベクトルを生成した。なお、平均プーリングはレビュー全体の平均的意味を知ること、最大プーリングは文に含まれる強い感情を読み取る目的がある。

7.3 極性判定モデルの構成

Sentence-BERTで生成したベクトルから極性を判定するモデルを作成した。今回使用したモデルの構成はMLPであり、その構成をFigure 7.1に示す。Figure 7.1のようにモデルの構成は複数の全結合層とReLU活性化関数により構成されており、段階的に抽象的な特徴を抽出する構造となっている。

Game Name	Reviews
MonsterHunter World	3,966件
MonsterHunter Rise	2,474件
MonsterHunter Wilds	3,990件
ドラゴンボール Sparking! ZERO	307件
ドラゴンボール ザ ブレイカーズ	590件
ドラゴンボール ファイターズ	68件
ドラゴンボール ゼノバース2	69件
UnderTale	2,374件
OverWatch2	6件
この赤いボタンを押さないで	4件
Apex Legends	32件
Doragon's Dogma 2	2,260件
Shadowverse: Worlds Beyond	13件
ストリートファイター6	3,923件
ELDEN RING	3,968件
Terraria	3,701件
Path of Exile 2	1,206件
Agony	14件
Rust	2,469件
Black Desert	96件
ARK: Survival Ascended	896件
サイバーパンク2077	3,028件
鉄拳8	689件

Table 7.1 Game Reviews.

7.4 学習

データセットはエポックごとにランダムにシャッフルを行い学習順による学習の偏りを防ぎ、その後バッチサイズを256としたミニバッチ学習を行った。バッチによるループは

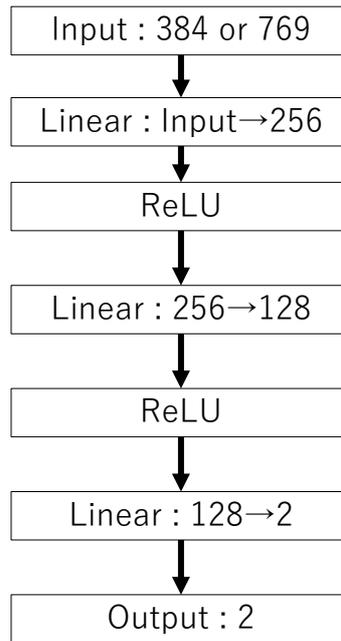


Figure 7.1 AI Model.

勾配をリセットし、順伝番による予測を行う。その後損失の計算を行い、逆伝播を行うことでどの重みが予測に悪影響があるかを調べ、勾配の更新を行うことでより良い予測結果を得られるようにした。損失関数には多クラス分類に有効である CrossEntropyLoss を、勾配の更新は安定した学習のと高速な収束を行うために Adam を用いた。

第8章 実験

AIの性能を評価するために学習回数毎に正答率、F1スコア、正答率とF1スコアの差を記録する。正答率は作成したAIがどれほど正確にレビュー文を評価できているか、F1スコアは適合率と再現率に大きな差があるか、正答率とF1スコアの差は学習データに偏りがあるかどうかを調べる目的がある。また、正答率は学習したデータの特徴をどれほど学習できているか、学習していないデータに対してどれほど適応できているかを調べるために学習に使用したデータと学習を行っていないデータの学習曲線を作成した。Figure 8.1、Figure 8.2、Figure 8.3はモデル解説の章で解説したAIモデルの正答率、F1スコア、正答率とF1スコアの差のグラフである。Figure 8.1より学習データに対しては学習する毎に正答率が上がり、未知のデータに対しては学習データほど正答率の変化がないものの学習回数が20を超えた時から正答率が0.02ほど下がったため学習回数20が過学習が起きた時であると考えられる。Figure 8.2よりF1スコアも正答率と同様に学習回数を重ねても0.8から0.02ほどの変化しかしないため適合率と再現率の変化も小さいことがわかる。Figure 8.3より正答率とF1の差が0.3を下回っているためデータの偏りがないことがわかる。

8.1 特徴の表現

レビューを「。、!、?」で区切り、それぞれの文をSentence-BERTによりベクトル化する。これらのベクトルを集約する際に、Sentence-BERT内のプーリングに倣い、平均プーリングと最大プーリングを用いた。したがって、本研究でのプーリングは、Sentence-BERT内のプーリングではなく、区切り毎のベクトルを集約するためのプーリングを指す。

テキストデータから感情を読み取る際、平均プーリングと最大プーリングのどちらを使えばよいかを調べるために、平均プーリングしたベクトル(384次元)、最大プーリングをしたベクトル(384次元)、両者を連結したベクトル(768次元)で学習を行った際どのような違いが生じるかを調べる。学習を行った結果特徴を平均化したデータの正答率、F1スコア、正答率とF1スコアの差をFigure 8.4、Figure 8.5、Figure 8.6に、特徴を最大化したデータの正答率、F1スコア、正答率とF1スコアの差をFigure 8.7、Figure 8.8、Figure 8.9に示す。Figure 8.1、Figure 8.4、Figure 8.7を比べると学習済みデータと学習を行っていないデータの双方において学習曲線に違いが現れないため、特徴の表現方法

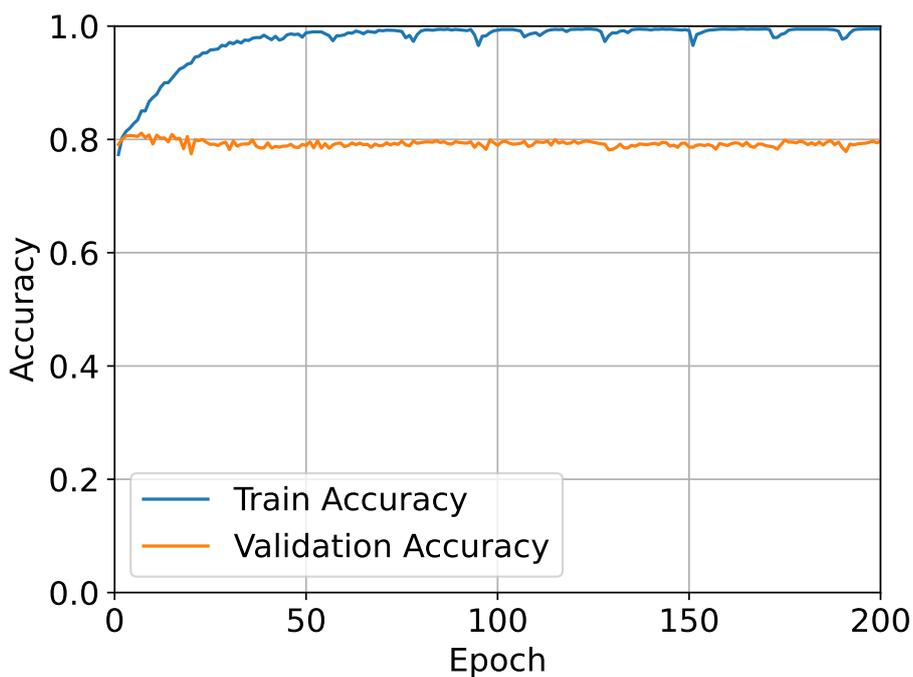


Figure 8.1 Standard Accuracy.

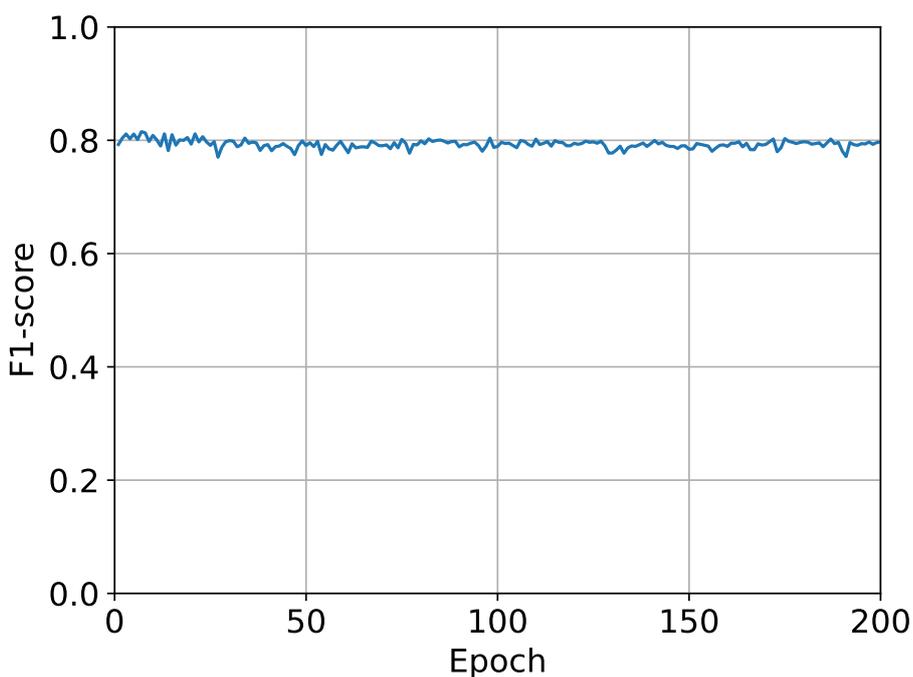


Figure 8.2 Standard F1.

を変えても正答率に影響しないことが分かった。Figure 8.2、Figure 8.5、Figure 8.8 を比べると正答率と同様に学習曲線に違いが現れないため特徴の表現方法を変えても適合率と再現率に変化が現れないことが分かった。Figure 8.3、Figure 8.6、Figure 8.9 を比べるとどのグラフも差の数値が0に近いので、特徴の表現により学習データの偏りは発

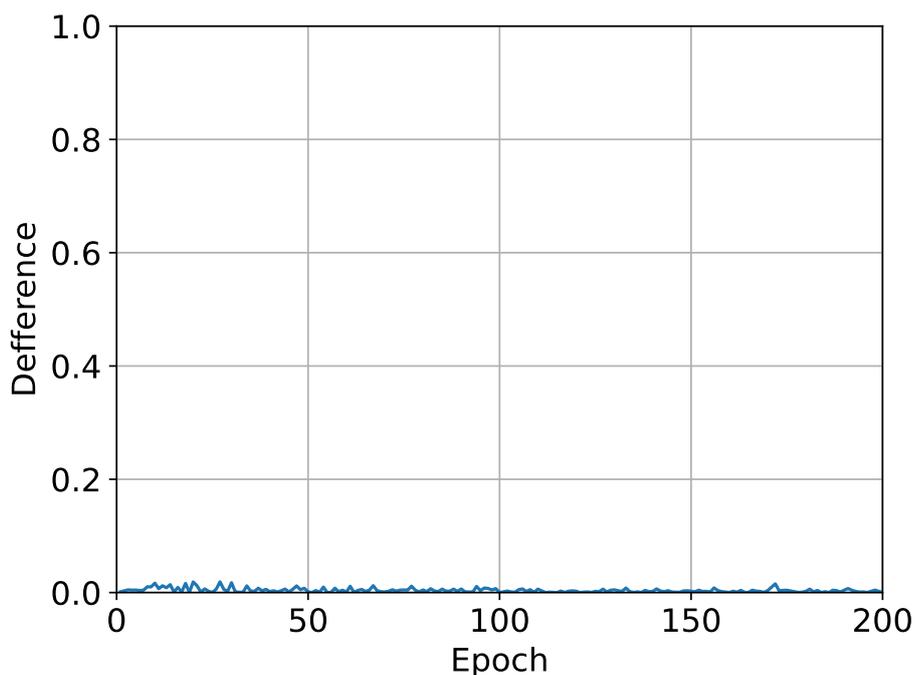


Figure 8.3 Standard Defference.

生しないことが分かった。このことから文ベクトルに変換する際に使用したプーリングの種類により AI の判定に変化は発生しないことが分かった。このような結果になった理由として、レビュー文は一文の中に肯定的な意見と否定的な意見を織り交ぜるのではなく、文ごとに否定的な意見と肯定的な意見を分けて書き込むため特徴の読み取り方を変えても数値のパターンに変化が発生しにくかったのではないかと考察する。

8.2 データ量の違い

AI には過学習という訓練データに対して過剰に適合し、未知のデータに対しては適切に予測できなくなる現象がある。その現象は学習するデータの量によっても起こりやすくなるのかを調べるためにデータ量を通常の学習用データセットである 10,000 件から否定的な意見と肯定的な意見が 500 になるように抽出した 1,000 件のデータと、学習用データ 10,000 件から否定的な意見と肯定的な意見が 500 になるように抽出した 100 件のデータで学習を行いグラフにする。学習用データが 1,000 件のデータで学習を行った結果の正答率、F1 スコア、正答率と F1 スコアの差のグラフを Figure 8.10、Figure 8.11、Figure 8.12 に、学習用データが 100 件のデータで学習を行った結果の正答率、F1 スコア、正答率と F1 スコアの差のグラフを Figure 8.13、Figure 8.14、Figure 8.15 に示す。Figure 8.1 と Figure 8.10、Figure 8.13 を比較すると学習に使用するデータの数が少なく

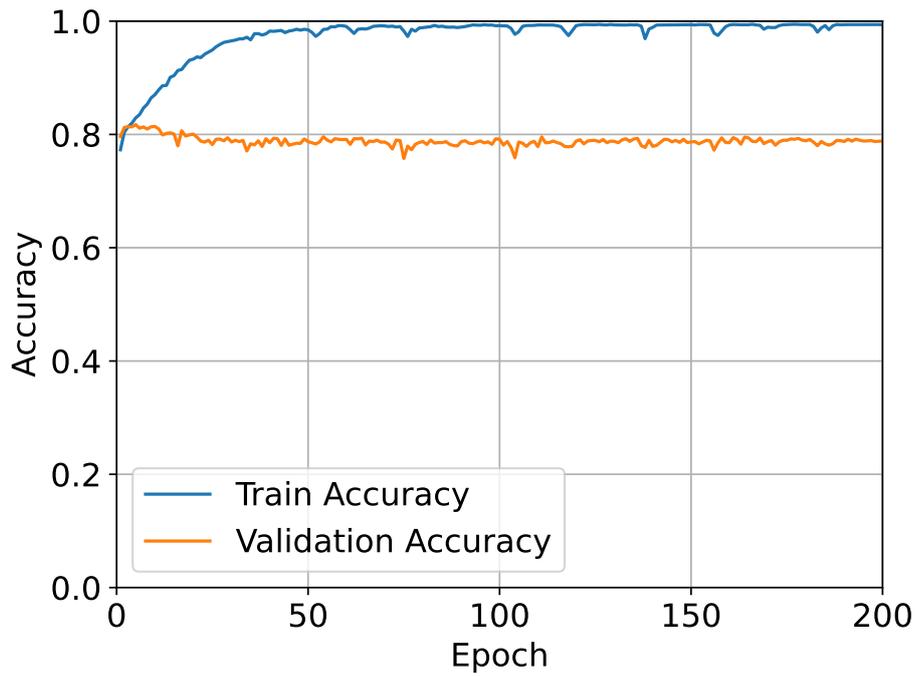


Figure 8.4 Accuracy mean.

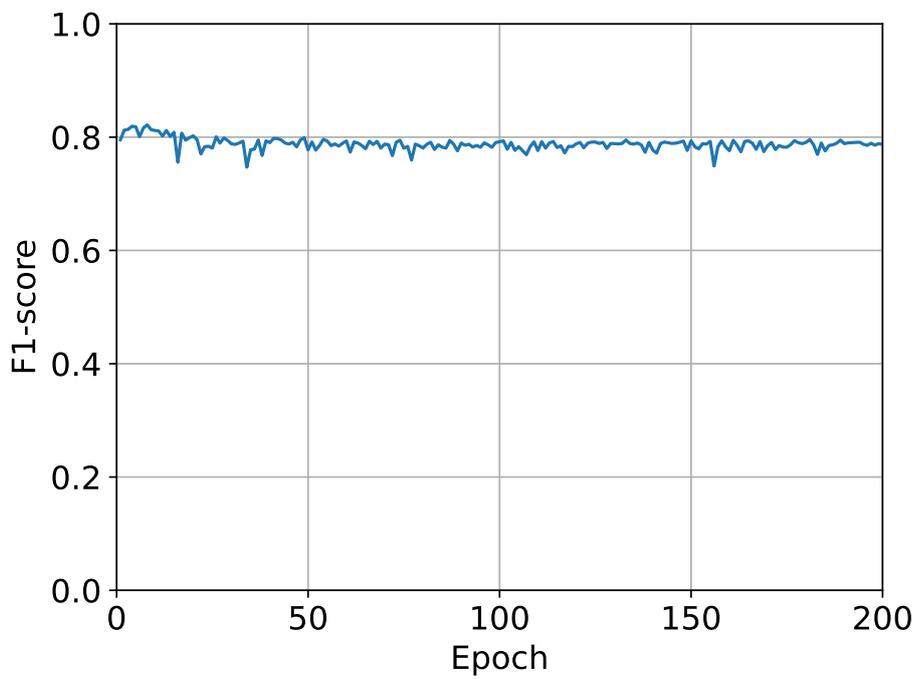


Figure 8.5 F1 mean.

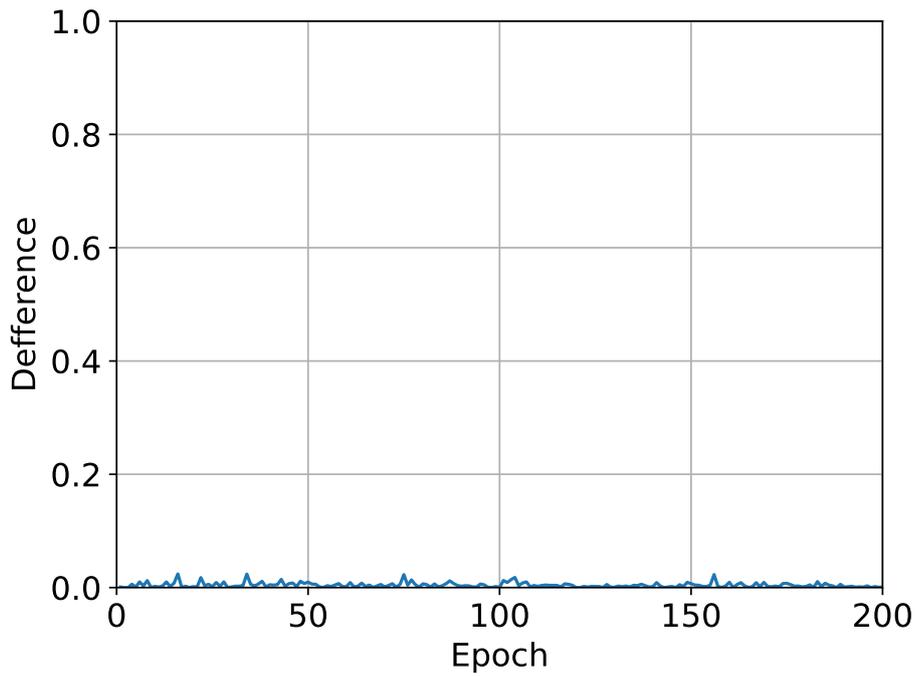


Figure 8.6 Defference mean.

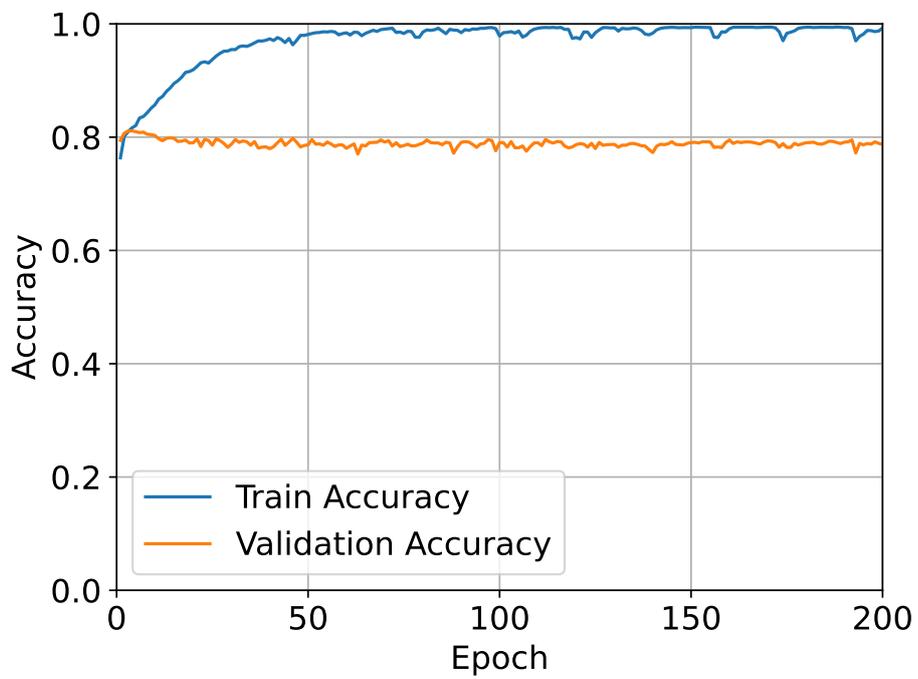


Figure 8.7 Accuracy max.

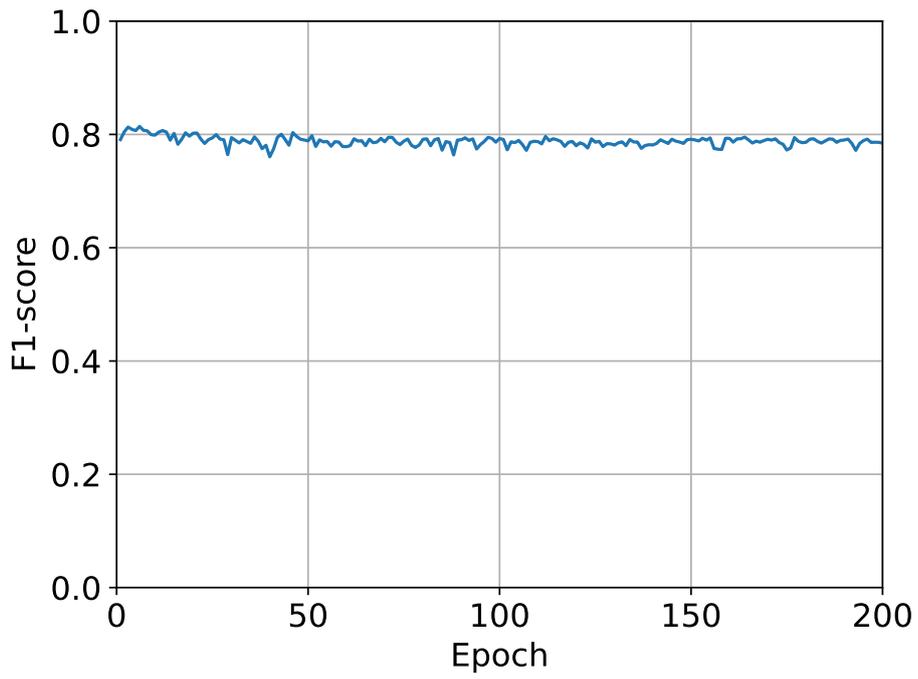


Figure 8.8 F1 max.

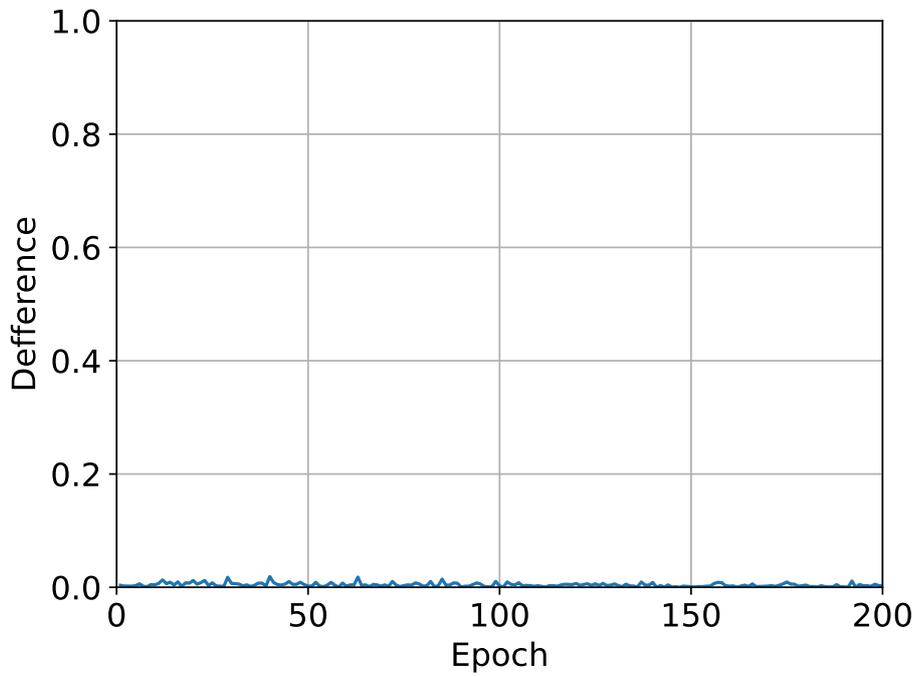


Figure 8.9 Defference max.

なるほど正答率が低くなること、学習済みデータの学習曲線はデータ数が多くなるほど滑らかになること、データ数が多くなるほど過学習が発生する学習回数が少なくなることがわかった。Figure 8.2 と Figure 8.11、Figure 8.14 を比較すると正答率と同様に学習データが少ないほど学習回数が1回の時のF1スコアの値が低くなること、学習に使用するデータの数が少なくなるほどF1スコアが低くなることがわかった。Figure 8.3 と Figure 8.12、Figure 8.15 を比較するとデータ数が100の時のみ学習回数が5回になるまで正答率とF1データの差が0.05であったため、五回学習を行うまではデータの偏りが発生していたことがわかる。このような結果になった理由として、データ数100の場合学習回数が5回を超えるまでデータの発生が発生していたこと、今回のモデルでは用意したデータを何度も反復学習をするモデルであるためデータ数が多いほど同じ特徴を持つデータが存在する確率が高くなり、テキストデータに存在する特徴をデータ数が少ない時より正確に学習できたのではないかと考察する。

8.3 データ量の差の解消

学習データを集める際、過学習を起こさないために学習データにばらつきがないようにする必要がある。しかし、データセットを作成する際できるだけ多くのデータを採集したいためデータの処理を変えてデータ量にばらつきがある状態でも正しく判別できる方法がないかと考えた。用意したデータセットは肯定的な意見と否定的な意見が同じになるように肯定的な意見を減らしたため、新たに肯定意見を減らさないデータを用意した。新たに用意したデータセットは学習に使用したデータは肯定的な意見が18,536件、否定的な意見が8,571件となった。なお、検証用のデータは正答率を正しく判別できるようにするため肯定的な意見と否定的な意見を両方2,857件にそろえた。この肯定的な意見と否定的な意見の差を判別できるようにするためには否定的な要素を読み取りやすくする必要があるという仮説を立て、文章内に「不十分、低い、退屈、単調、できない、悪い、薄れる、不親切、ストレス、バランスが悪い、萎える、緊張感が薄れる、自由度が低い」の単語を二重に登場させる。例えば「ゲームが悪い。」という文の場合「悪い」が含まれているため「ゲームが悪い悪い。」という文章に変換することで文章内の否定的な単語の影響を強めた。否定的な単語の影響を強めたAIの正答率、F1スコア、正答率とF1スコアの差をFigure 8.16、Figure 8.17、Figure 8.18に示す。Figure 8.1 と Figure 8.16 を比べると学習済みデータの学習曲線に違いは発生しなかったが、未学習データの分類について

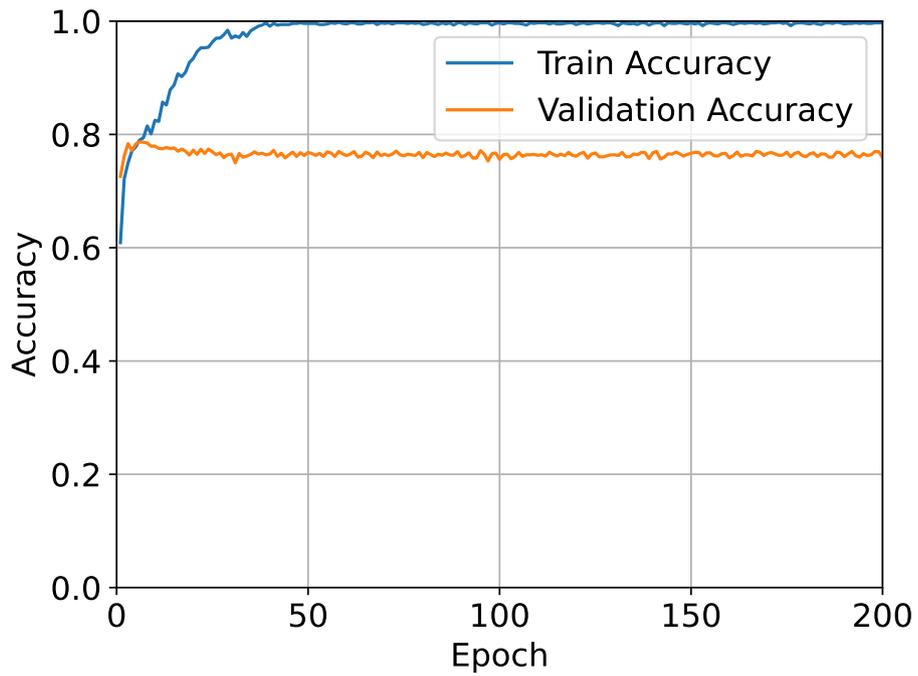


Figure 8.10 Accuracy 1000 data.

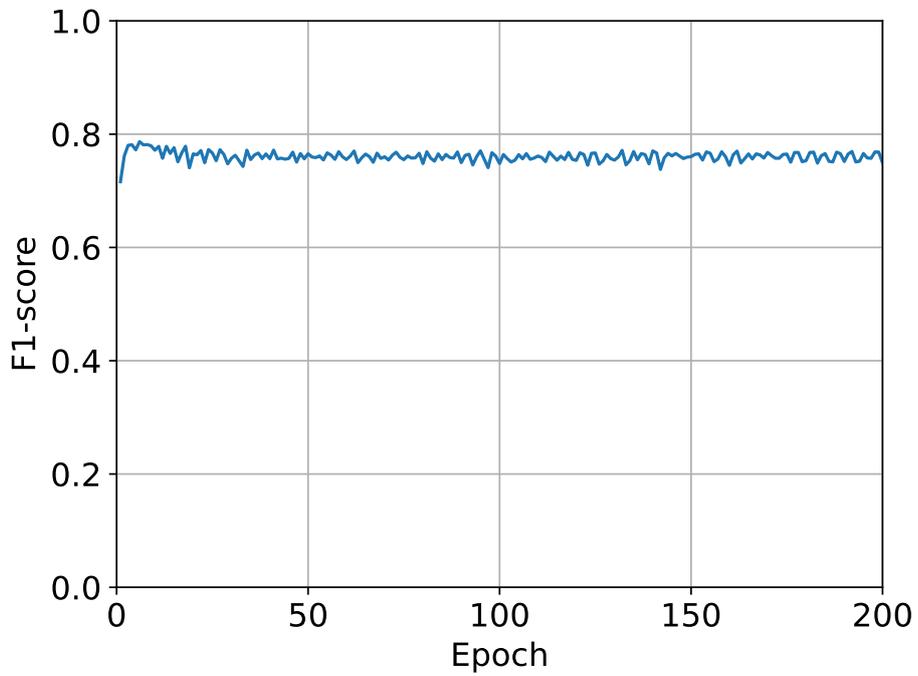


Figure 8.11 F1 1000 data.

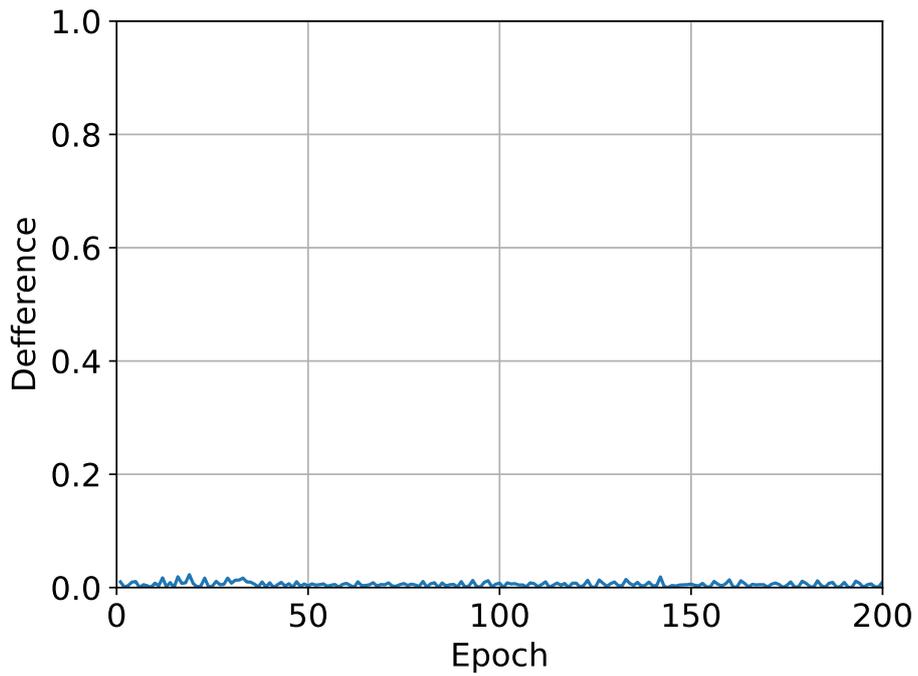


Figure 8.12 Defference 1000 data.

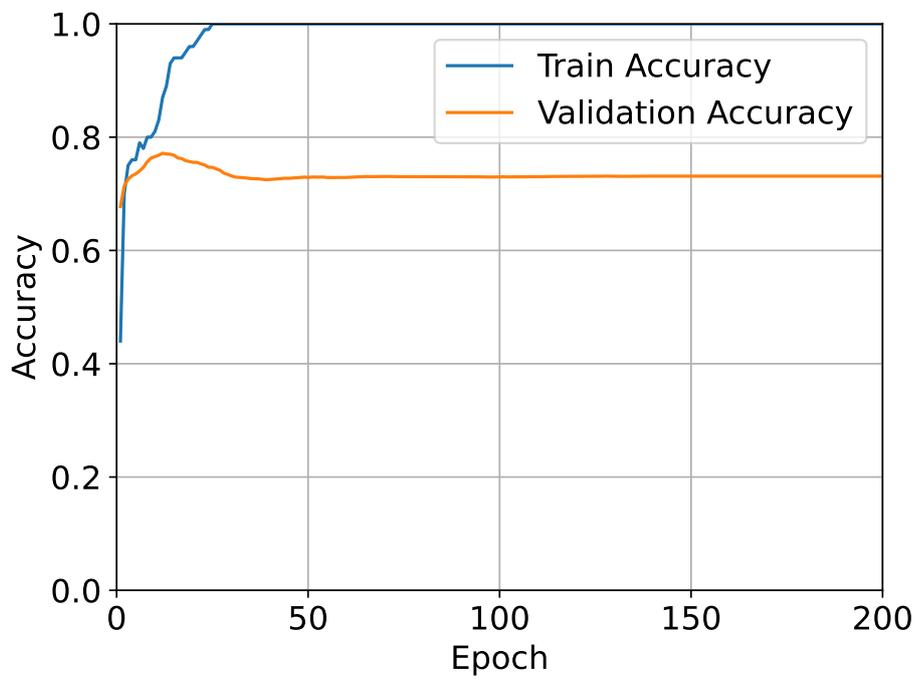


Figure 8.13 Accuracy 100 data.

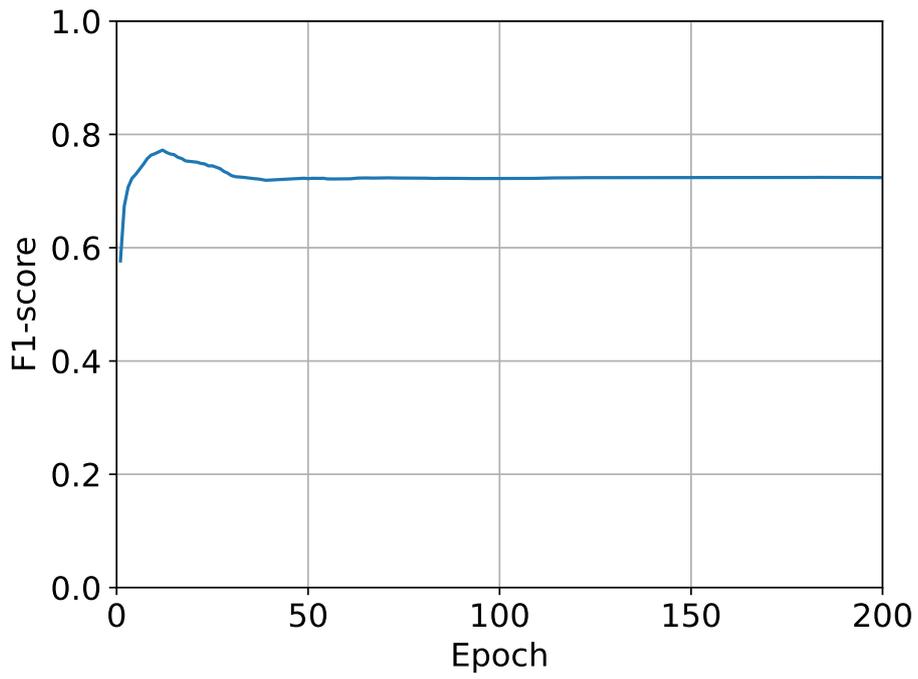


Figure 8.14 F1 100 data.

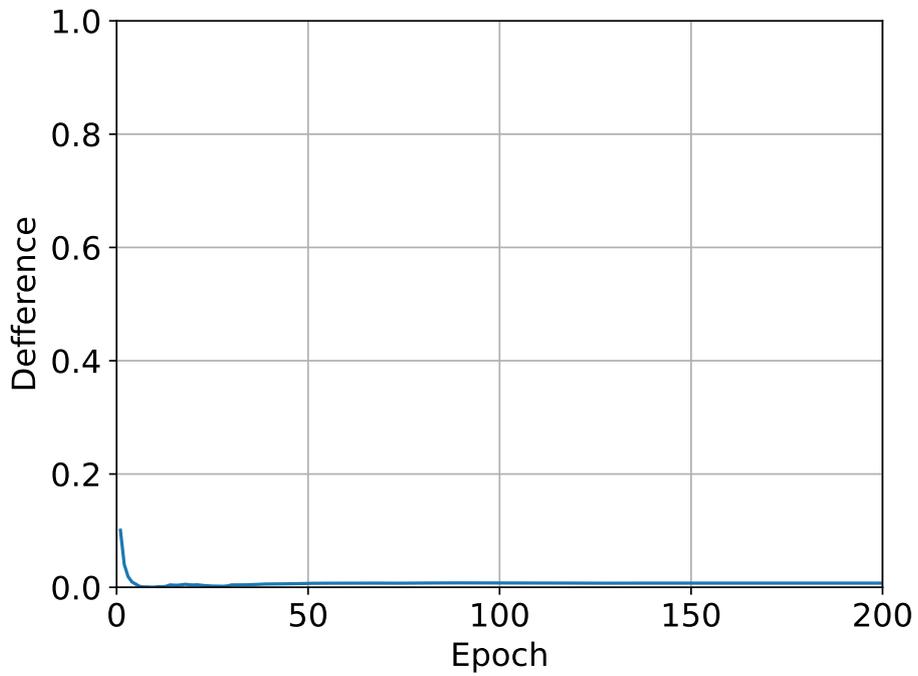


Figure 8.15 Defference 100 data.

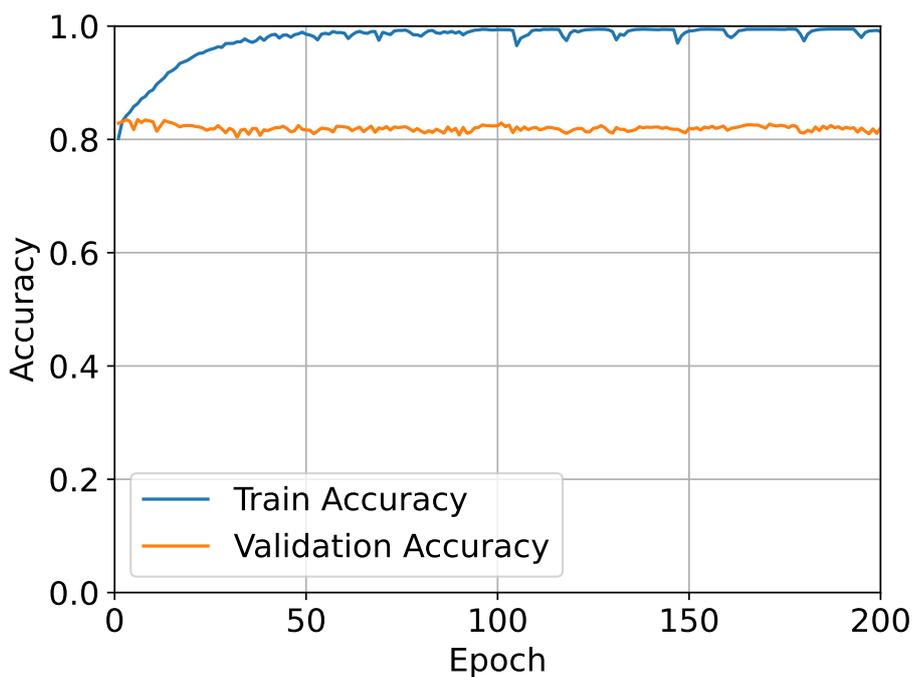


Figure 8.16 Accuracy data variability.

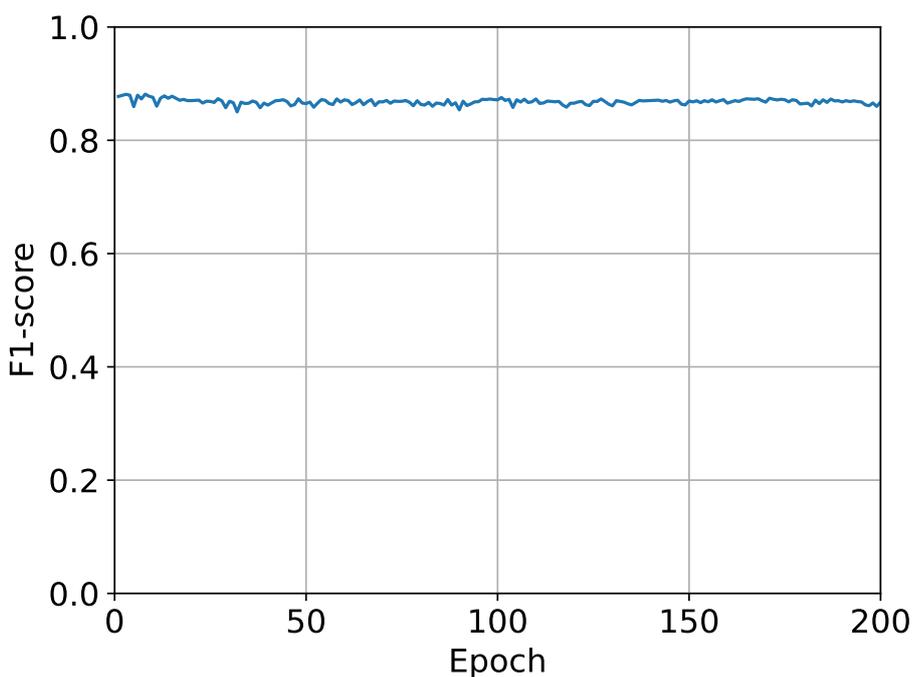


Figure 8.17 F1 data variability.

は否定的な単語の影響を強めた方がより良い正答率を得ることがわかった。Figure 8.2と Figure 8.17を比べると否定的な単語の影響を強めた方がF1スコアが0.09高くなり、適合率と再現率の差が縮まることがわかった。Figure 8.3と Figure 8.18を比べると Figure 8.3は差が0に近かったが図 8.18は差が0.05あるため肯定的な意見のほうが多いというデー

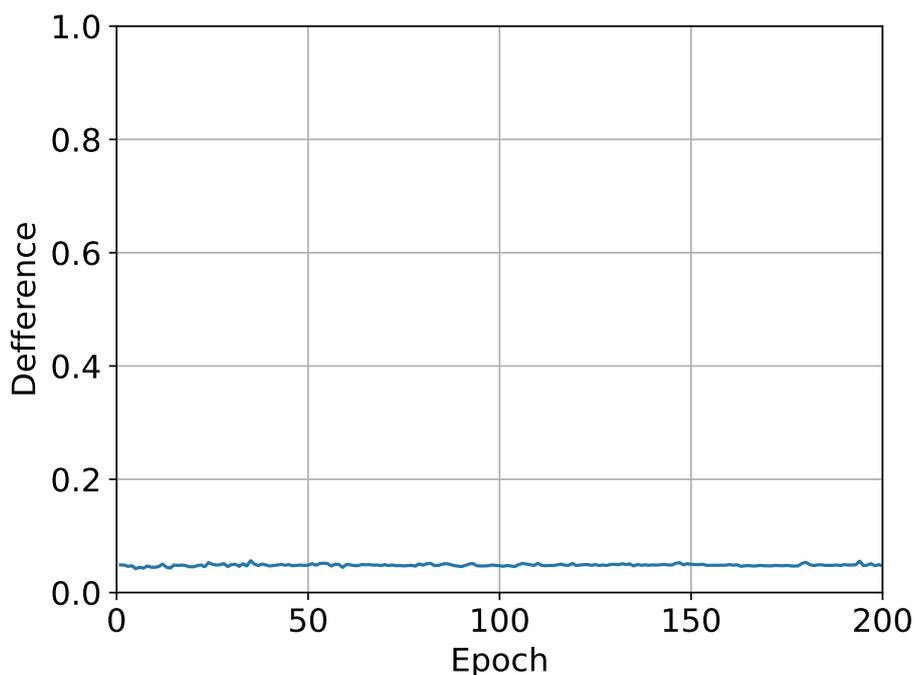


Figure 8.18 Defference data variability.

タの偏りは発生していることが分かった。このことからデータ量に偏りが発生する場合、肯定意見と否定意見の中で少数となる単語の影響を強めると分類の精度が向上することがわかった。このような結果となった理由としてデータを強調することで影響を強めた単語が示す否定意見のベクトル数値に関わらず指定した単語の分布や指定した単語が存在する量で判別する方式になり肯定意見と否定意見の特徴の違いがわかりやすくなったのではないかと考察する。

8.4 文章分割による違い

商品のレビューには長文や短文があり、言語処理を行う際長文のものは文章に含まれる文が短文より多くなる。レビュー文において文はポジティブな意見とネガティブな意見を区切るポイントになっているため文章を一つの文として処理するより文章を文に分解して処理した方が精度が高まると考えた。これを確認するため、新たに文章を「。、!、?」で区切らず Sentence-BERT に入力し、出力されたベクトルをそのまま用いるモデルを作成し、比較を行なった。文章を一つの文として考える場合の正答率、F1 スコア、正答率と F1 スコアの差を Figure 8.19、Figure 8.20、Figure 8.21 に示す。Figure 8.1 と Figure 8.19 を比べると学習済みデータの学習曲線に違いはないように見えるが、学習が済んでいないデータに関しては学習回数が 20 回を過ぎたときから正答率が下がり、最終的に文章を

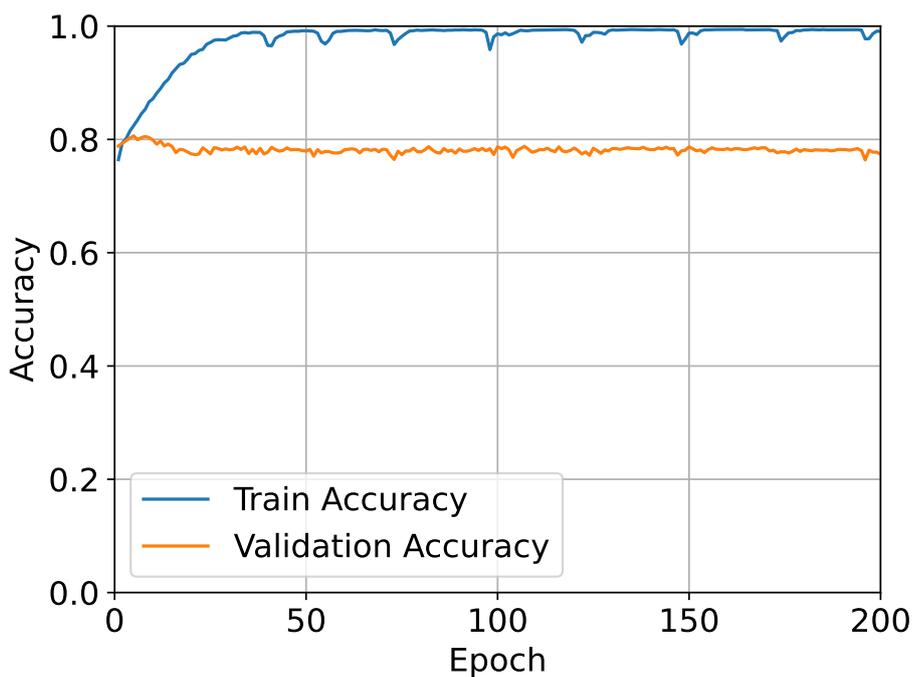


Figure 8.19 Accuracy all.

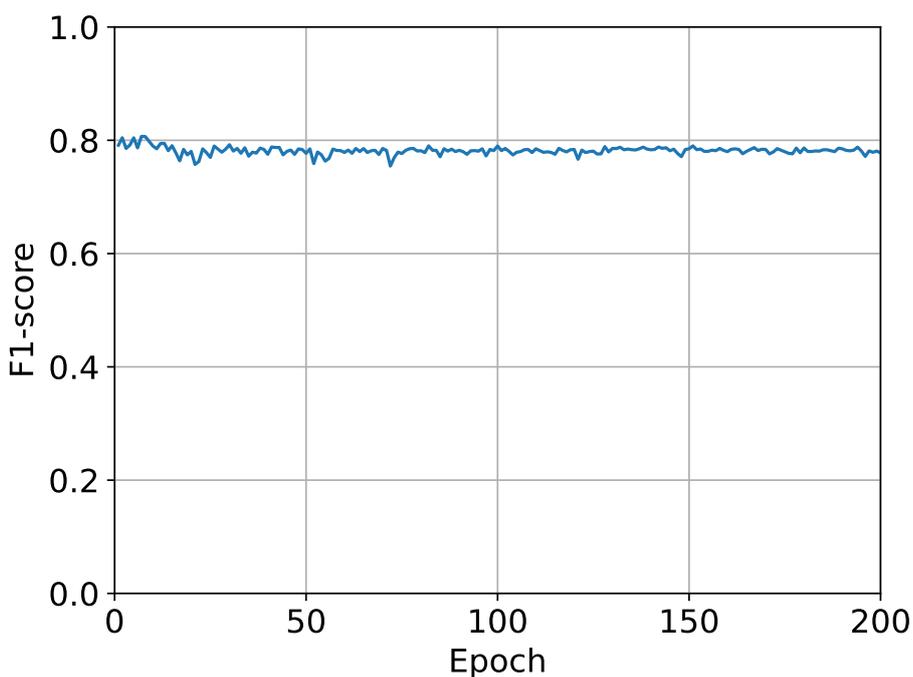


Figure 8.20 F1 all.

「。 , ! , ? 」で区切った場合の正答率より 0.02 低い確率となった。Figure 8.2 と Figure 8.20 を比べると正答率の学習が済んでいないデータと同様に学習回数が 20 回を過ぎたときから F1 スコアが下がり、最終的に 0.03 低い結果となった。Figure 8.3 と Figure 8.21 を比べると文章を文に分割しない場合でも学習データの偏りはないため「。 , ! , ? 」の区

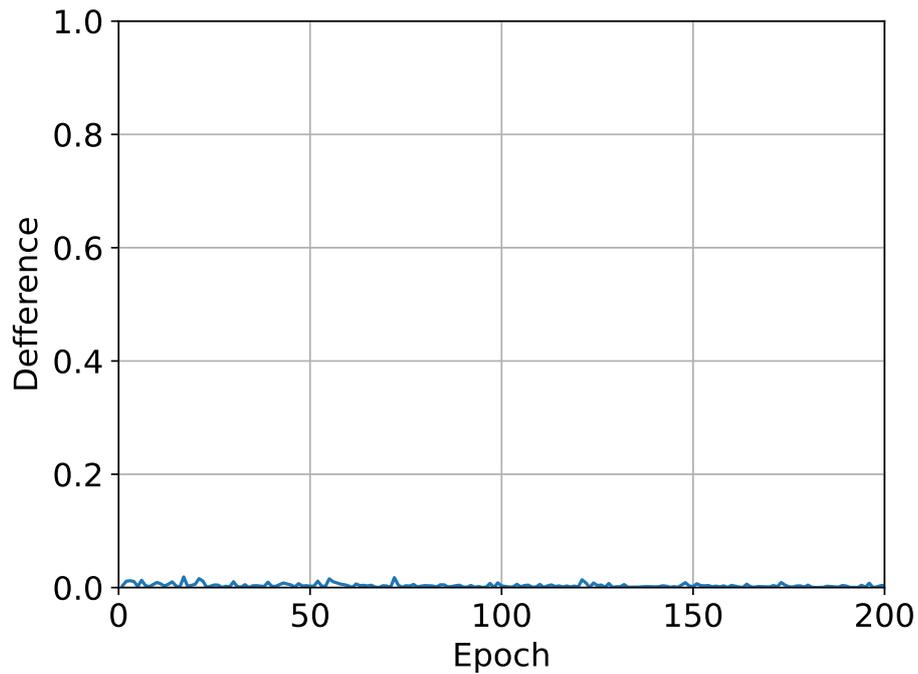


Figure 8.21 Defference all.

切りの有無により学習データの偏りは発生しないことが分かった。この結果から文章を「。!,?»で区切った場合のほうが良い学習結果となることが分かった。この結果となった理由として文の分割により学習データの偏りが発生しなかったことから文章が示す全体の評価に変化は起きにくいものの、文章を「。!,?»で区切ったことにより正答率とF1スコアが減少したことから「この点はよい。しかし、この点は悪く感じた。」のような文により区切られている意味の変化をより正確に判別できているのではないかと考えた。

第9章 結論

本研究では、Sentence-BERT を用いてベクトル化されたテキストデータの分類を行うニューラルネットワークの精度を高める方法を見つけるため、ゲームの商品レビューの極性判定を行う AI の学習方法を変化させた。実験の結果として、以下の4点が明らかとなった。

- (1) レビュー文では文単位で肯定意見と否定意見が記述される傾向があることから、プーリングの種類を変更しても分類精度に変化がなかったこと。
- (2) データ数が多いほど学習回数を少なくすることが必要であり過学習が起きる場合はデータ数を多くすると分類精度が向上すること。
- (3) データの肯定意見と否定意見に差が発生すると事前に判明している場合は、肯定意見か否定意見のどちらか一方に含まれる単語を事前に予測し、その単語の影響を強くすると肯定意見と否定意見の違いが分かりやすくなるため、データの肯定意見と否定意見を同じにするより良い分類結果を得られること。
- (4) 文章は文ごとにベクトル化を行うほうが文に含まれている否定意見と肯定意見をより正確に読み取れるため、分類制度が向上すること。

以上の点から、AI による分類を行う際はデータ数を多く、学習回数は少なくする方針が有効であり、事前に肯定意見と否定意見の量に違いが発生するとわかる場合は少数意見の特徴を強調する方式が有効である。

参考文献

- 1) 中外製薬株式会社, <https://www.chugai-pharm.co.jp/ptn/medicine/karada/karada022.html>, (閲覧日 2026年2月6日)
- 2) Sky株式会社, <https://www.skygroup.jp/media/article/3471/>, (閲覧日 2026年2月9日)
- 3) サイバネットシステム株式会社, https://www.cybernet.co.jp/ncs/column_glossary/glossary/gradient_descent/, (閲覧日 2026年1月20日)
- 4) 林栄一, <https://service.shiftinc.jp/ai-terminology/dropout/>, (閲覧日 2026年1月21日)
- 5) 三谷大暁, <https://ai-kenkyujo.com/artificial-intelligence/algorithm/neuralnetwork-activationfunction/>, (閲覧日 2026年1月21日)
- 6) 澁谷直樹, <https://note.com/kikaben/n/nddb38c635df5>, (閲覧日 2026年2月6日)
- 7) 株式会社日立ソリューションズ・クリエイト, <https://www.hitachi-solutions-create.co.jp/column/technology/text-mining.html>, (閲覧日 2026年1月22日)
- 8) 株式会社パーソルクロステクノロジー, https://staff.persol-xtech.co.jp/hatalabo/it_engineer/585.html, (閲覧日 2026年2月9日)
- 9) KDDI 株式会社, <https://biz.kddi.com/content/column/smartwork/what-is-textmining>, (閲覧日 2026年2月9日)
- 10) KDDI 株式会社, <https://biz.kddi.com/content/column/smartwork/what-is-textmining>, (閲覧日 2026年2月9日)
- 11) 株式会社 TechSword, <https://techsword.co.jp/column/how-to-evaluate-ai>, (閲覧日 2026年2月9日)
- 12) 株式会社アイスマイリー, https://aismiley.co.jp/ai_news/bert/, (閲覧日 2026年2月9日)
- 13) 株式会社オーグス総研, <https://www.ogis-ri.co.jp/otc/hiroba/technical/similar-document-search/part9.html>, (閲覧日 2026年2月9日)
- 14) 坂本将磨, <https://www.ai-souken.com/article/what-is-pytorch>, (閲覧日

2026年2月6日)